

243-02
174568

**1996
NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM**

**MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA**

**APPLYING PARALLEL PROCESSING TECHNIQUES TO
A TETHER DYNAMICS SIMULATION**

Prepared By: B. Earl Wells, Ph.D.

Academic Rank: Assistant Professor

Institution and Department: The University of Alabama in Huntsville
Department of Electrical and Computer Engineering

NASA/MSFC:
Office: Small Expendable Deployer Systems
Division: Earth and Space Science Projects
Branch: Science and Applications Projects

MSFC Colleagues: Jim Harrison
Pat Doty
Tommy Harris
Frayne Smith

The focus of this research has been to determine the effectiveness of applying parallel processing techniques to a sizable real-world problem, the simulation of the dynamics associated with a tether which connects two objects in low earth orbit, and to explore the degree to which the parallelization process can be automated through the creation of new software tools. The goal has been to utilize this specific application problem as a base to develop more generally applicable techniques.

The Tether Dynamics Problem

The basic tether deployment model, which served as a test-bed application for this research, is described in the NASA technical report outlined in reference [1] where it is assumed that two end bodies, each of point mass, are interconnected together via a nonconductive tether. It is also assumed that the tether is to be deployed by following a given deployment/control law and at least one of the end masses contains a finite amount of undepleted tether. The tether is assumed to be homogenous in composition, linearly elastic, and relatively thin, and the model also considers the longitudinal elasticity and bending stiffness.

Software Description

A comprehensive serial simulation of the tether dynamics problem has been developed by John Glaese of Control Dynamics Corporation. Throughout this research this simulation was used as the "gold" standard by which the accuracy of parallel versions of this simulation were measured. A general overview of the derivative function calculation portion (i.e. main loop) of this software is shown below in Figure 1. Approximately 800, lines of source code from the Control Dynamics simulation were converted into a Pseudo C syntax to be used by the automatic parallelization software.

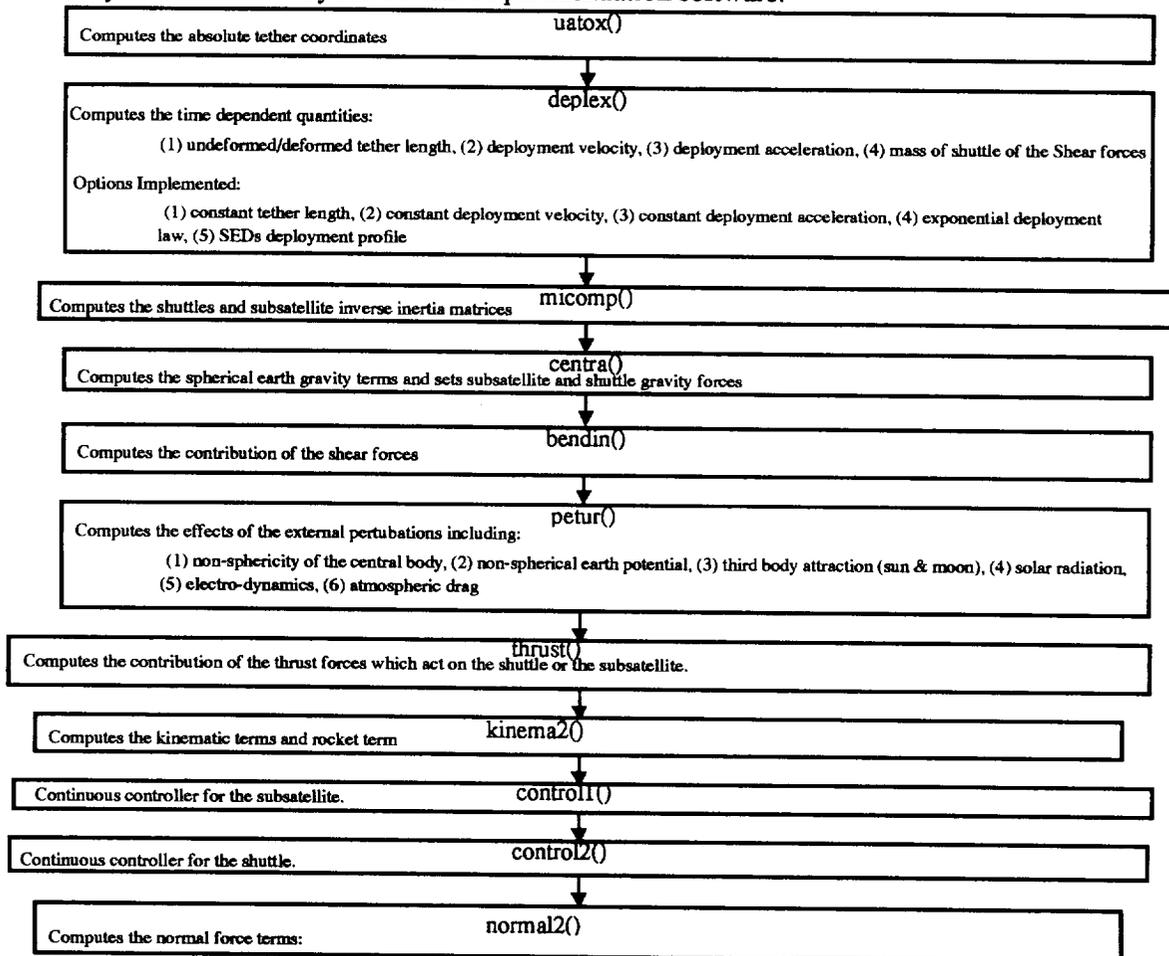


Figure 1: Overview of the Main Loop of the Tether Dynamics Simulation

Automatic Parallelization

To facilitate the parallelization of this type of problem on arbitrarily-connected multicomputing environments, a software tool called the *Automated Partitioning and Mapping Engine*, APME, was created. It functions in conjunction with a task allocator to automatically create effi-

cient allocations encoded through the source-level files that it produces. The APME software requires that the user enter two application-dependent input files, and it produces two types of source-level output files as shown in Figure 2. The input files are called the *Hardware Topology File* which describes the topology and hardware characteristics of the system, and the *Dynamic System Description File* which describe the structure of the simulation in CSSL form [2]. The output files include a set of *System Configuration Files* and a number of *High-Level-Language Files*. In this research, the FORTRAN representation of the tether dynamics problem was transformed manually into the required CSSL form. The format of these files is described in some detail in reference [2] with additional constructs being added such as the *\$mapping* section which describes how the problem is to be spatially mapped to the hardware which is described in the hardware topology file. In addition to the automatic decomposition of the problem along spacial domains the APME software also allows the selection of a number of parallel integration algorithms. At this point in the research, the APME software is approximately 70% complete.

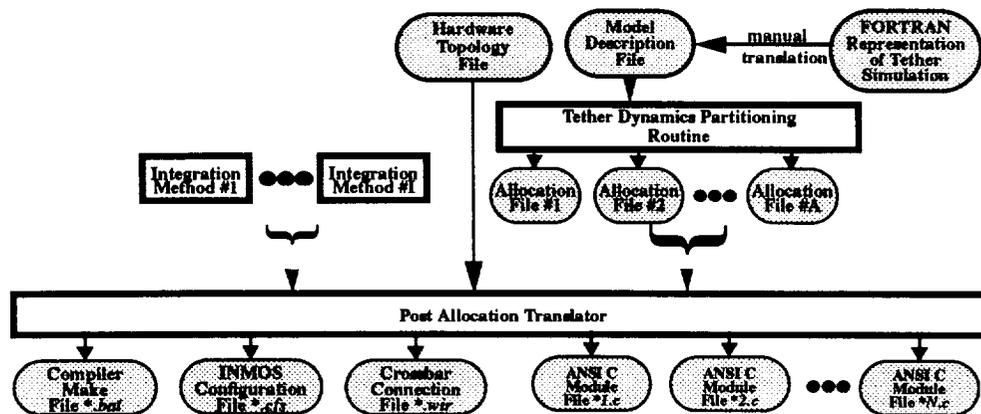


Figure 2: Proposed Automated Partitioning and Mapping Tool

Targeted Parallel Processing Environment

A small INMOS Transputer based system was used to determine the effectiveness of applying automatic parallelization to this general application domain. It served well in this capacity since it has many features that are present in other multicomputer architectures which have more powerful processors, while at the same time being a low-cost alternative to other commercially available parallel systems. Additionally, the Transputer system possesses a reconfigurable interconnection network which can be used to emulate a wide range of static topologies. This is accomplished, as shown in Figure 3, by superimposing the crossbar-connected links with the static links associated with the underlying fixed linear array topology present within the system. The base system had a total of ten T805 Transputers, with each Transputer having four bidirectional 20 Mb/s ports, two of which are used for the static connections and two of which are connected to the crossbar switch.

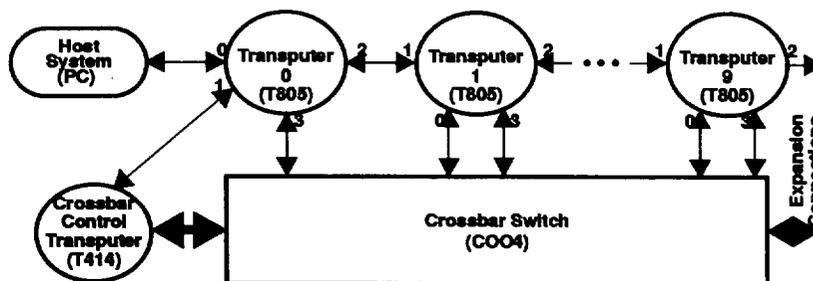
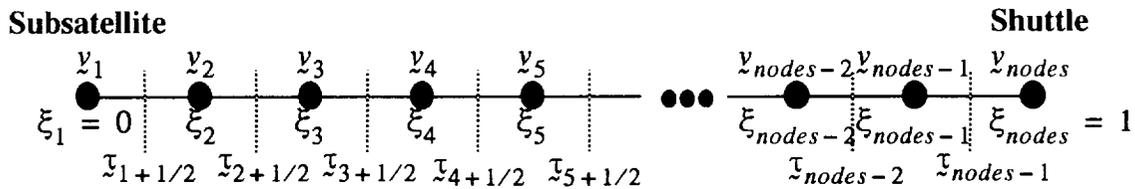


Figure 3: Targeted Parallel Processing Environment

Spatial Decomposition

The FORTRAN simulation of this model has been created by first transforming the equations of motion which describe the tether dynamics onto a new coordinate system which runs along the undeformed tether (it should be noted that transformations have been derived which map the underlying deformed tether model onto the more general deformed representation). This allows the set of partial differential equations to be converted into a set of ordinary differential equations by setting up a normalized coordinate system along the tether and discretizing at and exactly in between equally spaced (on the undeformed tether) nodal points. The resulting equations represent the individual nodal positions and velocities which given the position and velocity of one of the end bodies can be transformed to any viable coordinate system (see Figure 4).



Mapping Function: processor number = $\text{int}\left(\frac{\text{node number} - 1}{n}\right) + 1$

Figure 4: Spatial Decomposition of the Tether Dynamics Simulation

The underlying implications from a parallel processing point of view is that the amount of computation grows in proportion to the number of nodal points (i.e. the algorithm is $O(n)$ which is low-order polynomial) which in turn is somewhat proportional to the spacial resolution of the simulation. Additionally, the finite difference transformation results in a nearest and next-nearest neighbor communication pattern being present within the model if it is decomposed spacially along the undeformed tether; this maps well to a simple ring type topology which is highly desirable because of its low cost and high scalability.

Since the scope of the problem did not allow the modification of APME software to be completed by the end of the fellowship period, the parallel performance of this application was examined by profiling the APME-produced sequential simulation, and by evaluating the communication latency and bandwidth of the Transputer network. The following represents the results of this analysis under the following assumptions.

- 1) the mapping function of Figure 4 is used to place the nodal points on each Transputer,
- 2) the Crossbar Switch is statically configured into a virtual ring topology,
- 3) communication between processors is synchronous with no buffering of messages being allowed (blocking reads and blocking writes are the only communication mechanisms employed),
- 4) no single processor is allowed to simultaneously send and receive data to or from more than one other processor,
- 5) only simple store-and-forward routing of messages is supported,
- 6) cut through routing is not supported (if a communication must pass through an intermediate processor explicit communications must be scheduled on that processor to perform the echoing of the data through to the next processor along the communication path), and only ONE message can be sent through a link at a given time.

Table 1 highlights the results of this analysis for the arbitrarily chosen real-world case where it is assumed that a SEDS deployment scheme is employed, the model accounts for longitudinal deformations, the tether is inextensible, bending stiffness is neglected, nonsphericity is included, third body perturbations are neglected, atmospheric drag is included, diurnal and latitudinal effects are included and the radiation pressure is neglected. As shown in the second column of Table 1, during this analysis the time complexity for a portion of the simulation model was found to be linear with respect to the problem size but could not be parallelized without altering the algorithm. While it is believed that this section can be effectively parallelized, without significant loss of accuracy such investigations were reserved for future research.

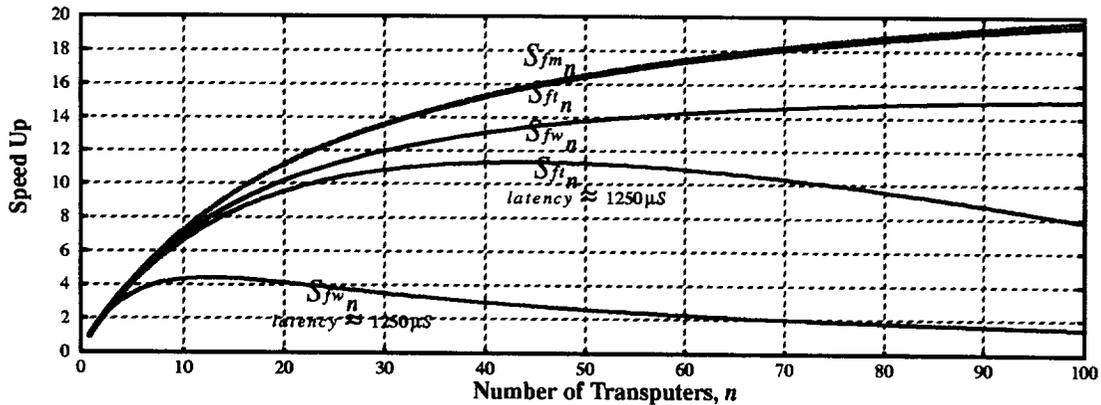
In addition to characterizing the time complexity of the workload, the table shows the type and cost of each communication which is required when the mapping function of Figure 4 is implemented. Given the aforementioned sequential portion of the code no isoefficiency function [3] can exist, implying that the efficiency will steadily decrease as the number of processors increases, but significant speedups can still be obtained. Figure 5 shows the fixed workload, fixed time, and fixed memory speedups [3], where it can be observed that the speedup associated with a fixed size problem of 200 nodal points reaches a maximum of 11 for a value of 42 processors, and the fixed time and fixed memory speedups both approach a value of 20 as the number of processors is increased from 1 to 100. It should be noted that applying new algorithmic techniques to the aforementioned sequential portion would result in all three speedups monotonically increasing as the number of processors increases in a linear manner making this problem highly scalable. It should also be noted that there exists a region from 1 to 10 processors in which the speedups are highly linear and efficiencies approach 80%. This region can be exploited at this time without changing the algorithm.

Figure 5 also shows the effects of communication channel latency on overall speedup. If the latency is set to a value which is considered typical of many parallel processing systems then severe performance degradation will be observed. This strengthens the case for the development of low latency environments, even if it comes with the price of decrease communication bandwidth.

Table 1: work load and communication time parameters -- spacial decomposition

Subroutine Name	Linear Coefficients		Constant Term (C)	Type of Collaborative Communication	Communication Delay for T805-30 System (ring topology)
	Serial (K_1)	Parallelizable (K_2)			
uatox	66	67	130	heterogeneous one-to-all	$C_T(2)(n-1)$
deplex	0	0	434	none	NA
micomp	0	0	230	none	NA
centra	0	158	128	none	NA
perturb	0	963	138	two, nearest neighbor	$2C_T(1)$
kinena2	0	130	60	four, nearest neighbor	$4C_T(1)$
control2	0	0	250	none	NA
normal2	0	127	79	four, nearest neighbor	$4C_T(1)$
uder	0	91	1552	homogeneous one-to-all	$C_T(4) \frac{(n+1)}{2}$
Total	66	1536	3001		

workload $\approx (K_1 + K_2) \text{ nodes} + C$ floating point operations $C_T(V) = \text{Transputer Communication Time} = 4.8V + 2.5 \text{ ms}$



Fixed Work Load Speed Up

$$S_{fw_n} = \frac{T_1(\text{nodes})}{T_n(\text{nodes})}$$

where,

nodes = 200 tether nodal points (fixed).

Fixed Time Speed Up

$$S_{ft_n} = \frac{T_1(\text{nodes})}{T_n(\text{nodes})}$$

where,

nodes set such that $T_1(200) \approx T_n(\text{nodes})$

Fixed Memory Speed Up

$$S_{fm_n} = \frac{T_1(\text{nodes})}{T_n(\text{nodes})}$$

where,

number of tether nodal points is proportional to n, (specifically, nodes = 200n)

Figure 5: Projected Speedups of Spacial Decomposition of Tether Simulation

Parallel Integration Algorithm

In addition to spacially decomposing the problem the underlying algorithm of the simulation, integration, can itself be parallelized. For example, it has been determined by a number of researchers that it is possible to create parallel versions of the basic predictor/corrector algorithms. One such set of such algorithms is the so-called Block Implicit Methods outlined in reference [4]. This method divides the simulation into blocks of time, where each block contains a set of equally spaced solution points. Parallelism is obtained by permitting the equations for each solution point within a block to be concurrently processed. With parallel block predictor-corrector algorithms (see Figure 6), each solution point results from the exe-

putation of one predictor and one corrector equation where all of the predictor equations associated with the solution points for a given block of time can be processed concurrently.

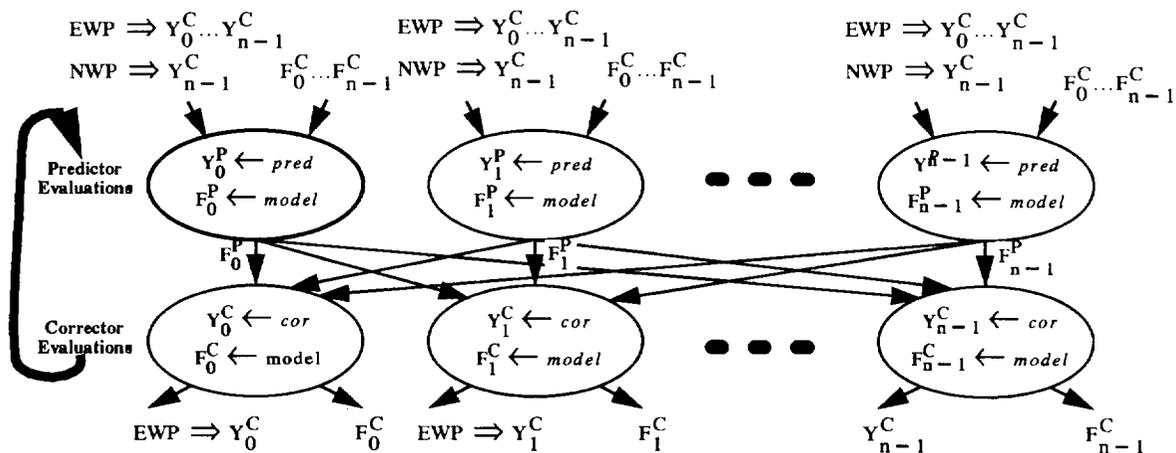


Figure 6: Structure of Parallel Block Predictor/Corrector Algorithm

As shown in Figure 7, the effects of applying the Null Weight Predictor/Corrector Algorithm to this problem shows very promising results for the case of 1 to 10 processors. It is important to note that the parallel version of the algorithm has similar accuracy characteristics when compared to its serial counterpart but has inferior stability. This decreased stability limits such algorithms applicability to implementations with 10 or few processors. It is possible to utilize parallel integration algorithms in combination with special decomposition techniques to have a multiplicative effect on overall performance. This is an area for future research.

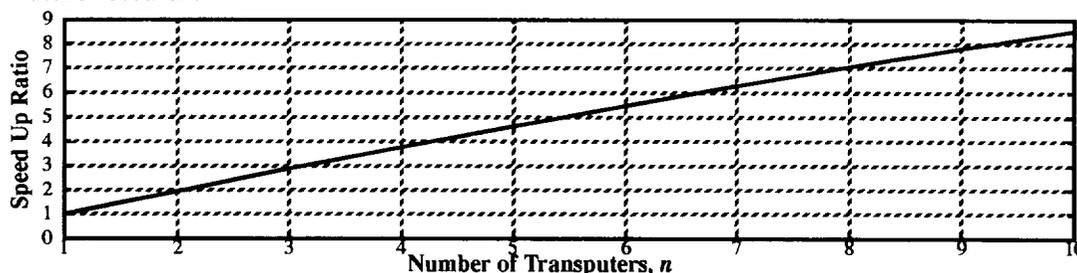


Figure 7: Projected Speed Up for Parallel NWP Block Predictor Corrector Algorithm

Conclusions

This research has shown that effective parallel representations can be obtained for sizable real-world problems such as this tether dynamics simulation using either special data domain decomposition techniques and/or parallel algorithmic techniques. This is important since it is expected that the simulation needs for these types of problems will continue to expand much faster than the raw computing power of our sequential computing resources as we explore much more complicated scenarios. Future research should be directed to develop new algorithmic techniques (both sequential and parallel), improve and develop new automated software tools such as the APME, and develop simulation paradigms which utilize existing supercomputing facilities and networked hardware resources such as PC/Workstation clusters (with the possible addition of low-latency communication hardware facilities).

References

1. P. Kohler, W. Maag, R. Wehrli, "Dynamics of a System of Two Satellites Connected by a Deployable and Extensible Tether of Finite Mass", NASA Technical Report, October 1978.
2. B. Earl Wells, "A Multicomputer Software Interface for Parallel Dynamic System Simulation", *Simulation Journal of the Society for Computer Simulation*, Vol. 65, No. 3, September 1995, pp. 191-205.
3. X. H. Sun and L. M. Ni, "Scalable Problems and Memory-Bound Speedup", *Journal of Parallel and Distributed Computing*, 1993.
4. L. G. Birta and O. Abou-Rabia, "Parallel Block Predictor-Corrector Methods for ODE's", *IEEE Transactions on Computers*, Vol. C-36, No. 3, March 1987, pp. 299-311.

